Bac technique

Page No.

info **I**

1 sp2020 2
2 sp2019 7
3 sp-c2019 11
4 sp2018 23
5 sp-c2018 27
6 sp2017 33
7 sp-c2017 36
8 sp-c2016 40
9 sp2015 45
10 sp-c2015 49

	Section : ..... **N° d'inscription :** ..... **Série :**	**Signature des surveillants**
	Nom et Prénom :	.....
	Date et lieu de naissance :	.....

✂ - - -

Ne rien écrire ici

Exercice 2 (*5 points*)

Soit la fonction **INCONNUE** suivante :

```
1) DEF FN INCONNUE (T :TAB ; p1,p2 :Entier) : Entier
2) Si (p1>p2)  Alors
       aux←p2
       p2←p1
       p1←aux
   Fin Si
3) S←0
   Pour i de p1 à p2  faire
     S← S+T[i]
   Fin Pour
4) INCONNUE ← S
5) Fin INCONNUE
```

N.B.
TAB = Tableau de 20 entiers

1) Nous proposons le tableau **T** suivant :

T	-2	19	-8	-14	4	5	-4	3	-8	9
	1	2	3	4	5	6	7	8	9	10

a. Remplir le tableau ci-dessous par la valeur de la variable **H** obtenue suite à l'exécution de l'instruction d'appel de la fonction **INCONNUE**.

N°	Instruction d'appel	Valeur de la variable H
1	**H ←INCONNUE** (T,3,8)	
2	**H ←INCONNUE** (T,6,2)	
3	**H ←INCONNUE** (T,4,4)	

b. Déduire le rôle de la fonction **INCONNUE.**

.....
.....
.....
.....
.....

Ne rien écrire ici

2) Nous appelons "**point d'équilibre**" d'un tableau **T** de taille **n** tout indice **k** qui sépare deux sous-tableaux ayant la même somme des éléments (**S1=S2**), avec **S1** est la somme des éléments indicés de **1** à **k-1** et **S2** est la somme des éléments indicés de **k+1** à **n**.

Exemple : pour le tableau **T** suivant, les points d'équilibre sont **4** et **7**.

T :	-2	19	-8	-14	4	5	-4	3	-8	9
	1	2	3	4	5	6	7	8	9	10

S1=9 **S2=9**

T :	-2	19	-8	-14	4	5	-4	3	-8	9
	1	2	3	4	5	6	7	8	9	10

S1=4 **S2=4**

En se basant sur la définition d'un point d'équilibre et en employant impérativement la fonction **INCONNUE**, compléter l'algorithme de la procédure **AFFICHE_EQUILIBRE** qui permet, pour un tableau **T** de **n** entiers donnés, de trouver et d'afficher tous les points d'équilibre s'ils existent. Le message ***"Aucun point d'équilibre"*** est à afficher dans le cas d'absence d'un point d'équilibre dans le tableau **T**.

0) DEF PROC AFFICHE_EQUILIBRE(T : TAB ; n : Entier)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Le sujet comporte 5 pages numérotées de 1/5 à 5/5.
Les pages 1/5, 2/5 et 3/5 sont à remplir par le candidat et à rendre avec sa copie.

Problème (10 points)

Un robot est capable de se déplacer dans un terrain rectangulaire en commençant d'un point de départ et en exécutant un parcours donné sous forme d'une succession de pas.

Les pas autorisés sont :

- Vers le haut, le pas est symbolisé par "**H**".
- Vers le bas, le pas est symbolisé par "**B**".
- Vers la droite, le pas est symbolisé par "**D**".
- Vers la gauche, le pas est symbolisé par "**G**".

Le terrain où se déplace le robot est représenté sous forme d'une grille formée de **NL** lignes et **NC** colonnes avec **NL** et **NC** sont deux entiers de l'intervalle **[1..100]**.

Initialement, le robot se trouve dans une position désignée par un numéro de ligne (**L**) et un numéro de colonne (**C**) où **L** est dans l'intervalle [1..**NL**] et **C** est dans l'intervalle [1..**NC**].

Le parcours est exprimé ainsi sous forme d'une séquence **S** formée uniquement par des lettres majuscules : "**H**", "**B**", "**G**" et "**D**".

Exemple : La séquence **S** = "**HHDDDGB**" décrit le parcours suivant : **2 pas vers le haut, 3 pas vers la droite, 1 pas vers la gauche et 1 pas vers le bas**.

Le robot exécute pas à pas le parcours décrit par la séquence **S** en commençant de sa position initiale de cordonnées **L** et **C** et il s'arrête dans l'un des deux cas suivants :

- **Le cas de dépassement** : Quand le dernier pas exécuté par le robot est à l'origine de sa sortie de la grille, un message est affiché "**Attention cas de dépassement !**" suivi par le numéro du pas qui est à l'origine du dépassement.
- **Le cas de fin du parcours** : Le robot a exécuté tous les pas de la séquence **S** sans aucun cas de dépassement. Un message est affiché contenant la nouvelle position du robot sur la grille.

Exemple :

Pour **NL** = **7** et **NC** = **9** et avec une position initiale du robot (**L** = **5**, **C** = **4**), nous présentons deux situations de parcours.

Situation1 (*parcours sur le schéma effectué avec des flèches vides* ⇨)

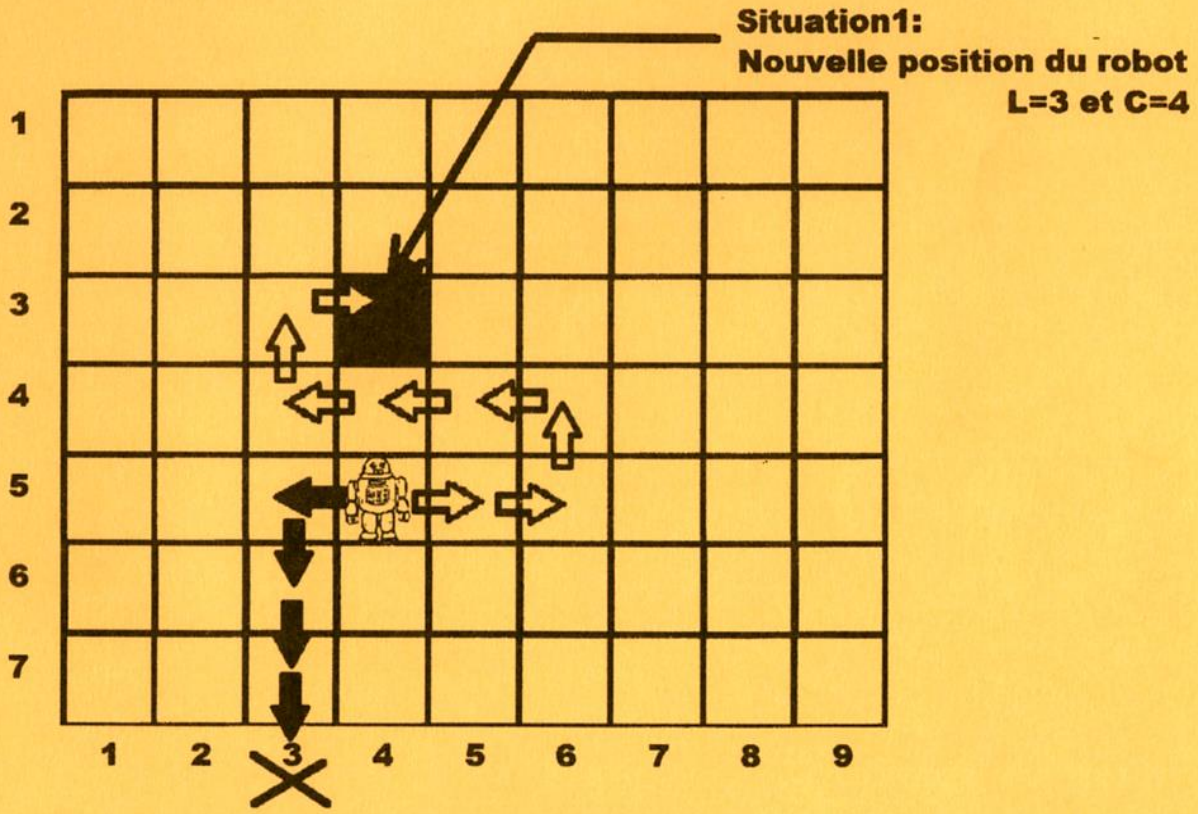
Pour la séquence **S = "DDHGGGHD"**

Le message affiché est : "Je suis à la position (3,4)"

Situation2 (*parcours sur le schéma effectué avec des flèches pleines* ➡)

Pour la séquence **S = "GBBBHDDDDGGHH"**

Le message affiché est : "Attention cas de dépassement causé par le pas N°4"



Situation2: Cas de dépassement

Nous proposons d'écrire un programme qui permet de simuler le parcours du robot. Pour se faire il s'agit :

- de saisir les dimensions de la grille **NL**, **NC** où se déplace le robot, sa position initiale **L** et **C** et la séquence de parcours à suivre **S**.
- d'afficher le résultat final du parcours selon le procédé décrit précédemment.

Travail demandé :

1) Analyser le problème en le décomposant en modules.
2) Ecrire les algorithmes des modules envisagés et présenter les tableaux de déclaration des objets relatifs à ces modules.

Section : ..... N° d'inscription : ..... Série : .....

Nom et Prénom : .....

Date et lieu de naissance : .....

Signature des surveillants

.....

.....

*Épreuve : **INFORMATIQUE***

*Sections: **Mathématiques, Sciences expérimentales et Sciences techniques***

*Session **2019***

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.

Les pages 1/4 et 2/4 sont à remplir par le candidat et à rendre avec sa copie.

Exercice 1 (*4 points*)

Dans un contexte informatique et pour chacune des propositions ci-dessous, mettre dans la case correspondante la lettre "**V**" si la proposition est correcte ou la lettre "**F**" dans le cas contraire.

1) Soit l'algorithme suivant :

```
0) Début Inconnu
1) Lire (C1)
2) Lire (C2)
3) C3 ← 0
   Pour i de 1 à Long(C2) faire
     Si (Majus(C2[i]) = Majus(C1))
        Alors C3 ← C3 + 1
     Fin si
   Fin pour
4) Ecrire (C3)
5) Fin Inconnu
```

a) Ci-dessous des extraits de propositions de tableaux de déclaration des objets utilisés. La déclaration correspondante à l'algorithme **Inconnu** est :

Objet	Type/nature
C1	Chaîne
C2	Chaîne
C3	Entier

☐

Objet	Type/nature
C1	Caractère
C2	Caractère
C3	Entier

☐

Objet	Type/nature
C1	Chaîne
C2	Caractère
C3	Réel

☐

Objet	Type/nature
C1	Caractère
C2	Chaîne
C3	Entier

☐

b) Afin d'améliorer le message d'affichage du résultat de l'algorithme précédent et de le rendre significatif relativement au traitement effectué, l'instruction numéro **4** sera remplacée par l'instruction suivante :

☐ Ecrire ("Le nombre de caractères majuscules de ",C1," et ",C2," est : ",C3)

☐ Ecrire ("Le nombre d'occurrences de ",C1," dans ",C2," est : ",C3)

☐ Ecrire ("Le nombre de chiffres dans ",C2," est : ",C3)

☐ Ecrire ("Le nombre de caractères communs entre ",C1," et ",C2," est : ",C3)

Ne rien écrire ici

2) Soit la suite **U** définie par $\begin{cases} U_0 = 1 \\ U_n = 1+1/U_{n-1} \quad \text{pour tout } n > 0 \end{cases}$

a) La séquence algorithmique qui permet de déterminer le terme **Un** avec $n \geq 0$ est :

T[1] ← 1
Pour i de 2 à n+1 faire
  T[i] ← 1+1/T[i-1]
Fin pour
Un ← T[n+1]
	U0 ← 1
	Pour i de 1 à n faire
	  Un ← 1+1/U0
	Fin pour
		Un ← 1
		Pour i de 1 à n faire
		  Un ← 1+1/Un
		Fin pour
			U0 ← 1
			Pour i de 1 à n faire
			  Un ← 1+1/U0
			  U0 ← Un
			Fin pour
			Un ← U0

☐	☐	☐	☐

b) L'entête de la fonction qui permet de déterminer le terme **Un** avec $n \geq 0$ est :

DEF FN Suite (n : Réel) : Réel	☐	DEF FN Suite (n : Entier) : Entier	☐
DEF FN Suite (n : Entier) : Réel	☐	DEF FN Suite (n : Réel) : Entier	☐

Exercice 2 (*3 points*)

Soit la séquence algorithmique ci-dessous, où **x** est un entier naturel :

```
nb ← 1
Tant que (x Div 10 ≠ 0) faire
   nb ← nb + 1
   x ← x Div 10
Fin tant que
```

1) Compléter le tableau ci-contre par la valeur finale de **nb** suite à l'exécution de cette séquence pour chacune des valeurs de **x**.

x	nb
5403	
176	
3	

2) Donner le rôle de cette séquence.

3) Ecrire une séquence algorithmique équivalente à celle donnée précédemment sans utiliser une structure itérative.

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION EXAMEN DU BACCALAURÉAT	Session 2019	
	Épreuve : INFORMATIQUE	Sections : Mathématiques, Sciences expérimentales et Sciences techniques
	Durée : 1h 30	Coefficient de l'épreuve : 0.5

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
Les pages 1/4 et 2/4 sont à remplir par le candidat et à rendre avec sa copie.

Problème (*13 points*)

Une société commerciale cherche à automatiser l'accès à ses services via des cartes à puces. Chaque carte est caractérisée par un identifiant unique, une date de création et un code d'accès. Pour cela, on propose d'écrire un programme permettant de :

1) Stocker les identifiants de **N** cartes dans un tableau **IDENT** (avec $3 \leq N \leq 50$), sachant que l'identifiant est une chaîne formée de **8** chiffres,
2) Stocker les dates de création de ces **N** cartes d'accès dans un tableau **DATE**, sachant qu'une date de création est une chaîne de caractères formée de deux parties : le jour **J** et le mois **M** séparés par le caractère "/" (On ne tiendra compte que des contrôles suivants : **J** est un entier compris entre 1 et 31 et **M** est un entier compris entre 1 et 12).
3) Déterminer dans un tableau **CODE** les **N** codes à partir du tableau **DATE** en utilisant le principe ci-après, sachant qu'un code est une chaîne de **4** chiffres :
   - Concaténer **J** et **M** pour former un nombre de **2**, **3** ou **4** chiffres.
   - Multiplier ce nombre par un entier aléatoire **X** compris entre **5** et **64** afin d'obtenir un nouveau nombre **Y**.
   - Former le code de quatre chiffres comme suit :
     - Si **Y** est un entier de **4** chiffres, déplacer le chiffre des milliers vers la droite de ce nombre.
     - Si **Y** est inférieur à 1000, ajouter des zéros (**0**) à droite de ce nombre pour que sa longueur soit égale à **4**.
     - Si **Y** est supérieur à 9999, additionner le nombre formé des 4 chiffres de gauche au nombre formé des chiffres restants à droite jusqu'à ce que sa longueur soit égale à **4**.
4) Afficher les informations relatives à chacune des cartes sous la forme suivante :

Identifiant de la carte : Date de sa création correspondante : Code correspondant

NB : Le candidat n'est pas appelé à vérifier l'unicité des identifiants dans le tableau **IDENT**.

Exemple :

Pour **N**=3 et les deux tableaux **IDENT** et **DATE** suivants :

IDENT	12345678	23456789	34567891
	1	2	3
DATE	8/11	2/1	24/12
	1	2	3

❖ On obtient le tableau **CODE** suivant :

CODE	1108	6720	1314
	1	2	3

En effet, les codes des cartes sont obtenus comme suit :

- Pour la carte **N°1**, la concaténation de **J** et de **M** donne le nombre **811**. En supposant que l'entier aléatoire **X = 10**, le nombre obtenu est égal à **8110** (811 * 10) qui est composé de 4 chiffres. En déplaçant le chiffre des milliers (**8**) vers la droite, on obtient le code **1108**.
- Pour la carte **N°2**, la concaténation de **J** et de **M** donne le nombre **21**. En supposant que l'entier aléatoire **X = 32**, le nombre obtenu est égal à **672** (21 * 32) qui est inférieur à 1000. En ajoutant un zéro à droite pour que sa longueur soit égale à **4**, on obtient le code **6720**.
- Pour la carte **N°3**, la concaténation de **J** et de **M** donne le nombre **2412**. En supposant que l'entier aléatoire **X = 53**, le nombre obtenu est égal à **127836** (2412 * 53) qui est supérieur à 9999.

  Le nombre formé des 4 chiffres de gauche (**1278**) sera additionné au nombre formé des chiffres restants à droite (**36**), on obtient le code **1314** (1278 + 36).

❖ Le programme affiche :

12345678 : 8/11 : 1108

23456789 : 2/1 : 6720

34567891 : 24/12 : 1314

Travail demandé :

1) Analyser le problème en le décomposant en modules.
2) Ecrire les algorithmes et les tableaux de déclaration des objets relatifs aux modules envisagés.

Section : ..... **N° d'inscription :** ..... **Série :** .....

Nom et Prénom : .....

Date et lieu de naissance : .....

Signature des surveillants

.....

.....

Épreuve : ***INFORMATIQUE***

Sections: ***Mathématiques, Sciences expérimentales et Sciences techniques***

Session ***2019***

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.

Les pages 1/4 et 2/4 sont à remplir par le candidat et à rendre avec sa copie.

Exercice 1 (*4 points*)

Dans un contexte informatique et pour chacune des propositions ci-dessous, mettre dans la case correspondante la lettre "**V**" si la proposition est correcte ou la lettre "**F**" dans le cas contraire.

1) Soit l'algorithme suivant :

```
0) Début Inconnu
1) Lire (C1)
2) Lire (C2)
3) C3 ← 0
   Pour i de 1 à Long(C2) faire
     Si (Majus(C2[i]) = Majus(C1))
        Alors C3 ← C3 + 1
     Fin si
   Fin pour
4) Ecrire (C3)
5) Fin Inconnu
```

a) Ci-dessous des extraits de propositions de tableaux de déclaration des objets utilisés. La déclaration correspondante à l'algorithme **Inconnu** est :

Objet	Type/nature
C1	Chaîne
C2	Chaîne
C3	Entier

☐

Objet	Type/nature
C1	Caractère
C2	Caractère
C3	Entier

☐

Objet	Type/nature
C1	Chaîne
C2	Caractère
C3	Réel

☐

Objet	Type/nature
C1	Caractère
C2	Chaîne
C3	Entier

☐

b) Afin d'améliorer le message d'affichage du résultat de l'algorithme précédent et de le rendre significatif relativement au traitement effectué, l'instruction numéro **4** sera remplacée par l'instruction suivante :

☐ Ecrire ("Le nombre de caractères majuscules de ",C1," et ",C2," est : ",C3)

☐ Ecrire ("Le nombre d'occurrences de ",C1," dans ",C2," est : ",C3)

☐ Ecrire ("Le nombre de chiffres dans ",C2," est : ",C3)

☐ Ecrire ("Le nombre de caractères communs entre ",C1," et ",C2," est : ",C3)

2) Soit la suite **U** définie par $\begin{cases} U_0 = 1 \\ U_n = 1+1/U_{n-1} \quad \text{pour tout } n > 0 \end{cases}$

a) La séquence algorithmique qui permet de déterminer le terme **Un** avec **n ≥ 0** est :

T[1] ← 1
Pour i de 2 à n+1 faire
  T[i] ← 1+1/T[i-1]
Fin pour
Un ← T[n+1]
	U0 ← 1
	Pour i de 1 à n faire
	  Un ← 1+1/U0
	Fin pour
		Un ← 1
		Pour i de 1 à n faire
		  Un ← 1+1/Un
		Fin pour
			U0 ← 1
			Pour i de 1 à n faire
			  Un ← 1+1/U0
			  U0 ← Un
			Fin pour
			Un ← U0

☐	☐	☐	☐

b) L'entête de la fonction qui permet de déterminer le terme **Un** avec **n ≥ 0** est :

DEF FN Suite (n : Réel) : Réel	☐	DEF FN Suite (n : Entier) : Entier	☐
DEF FN Suite (n : Entier) : Réel	☐	DEF FN Suite (n : Réel) : Entier	☐

Exercice 2 (*3 points*)

Soit la séquence algorithmique ci-dessous, où **x** est un entier naturel :

```
nb ← 1
Tant que (x Div 10 ≠ 0) faire
    nb ← nb + 1
    x ← x Div 10
Fin tant que
```

1) Compléter le tableau ci-contre par la valeur finale de **nb** suite à l'exécution de cette séquence pour chacune des valeurs de **x**.

x	**nb**
5403	
176	
3	

2) Donner le rôle de cette séquence.

3) Ecrire une séquence algorithmique équivalente à celle donnée précédemment sans utiliser une structure itérative.

RÉPUBLIQUE TUNISIENNE **MINISTÈRE DE L'ÉDUCATION** **EXAMEN DU BACCALAURÉAT**	**Session 2019**	
	Épreuve : **INFORMATIQUE**	**Sections :** **Mathématiques,** **Sciences expérimentales et Sciences techniques**
	Durée : **1h 30**	Coefficient de l'épreuve : **0.5**

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
Les pages 1/4 et 2/4 sont à remplir par le candidat et à rendre avec sa copie.

Problème (*13 points*)

Une société commerciale cherche à automatiser l'accès à ses services via des cartes à puces. Chaque carte est caractérisée par un identifiant unique, une date de création et un code d'accès. Pour cela, on propose d'écrire un programme permettant de :

1. Stocker les identifiants de **N** cartes dans un tableau **IDENT** (avec $3 \leq \mathbf{N} \leq 50$), sachant que l'identifiant est une chaîne formée de **8** chiffres,
2. Stocker les dates de création de ces **N** cartes d'accès dans un tableau **DATE**, sachant qu'une date de création est une chaîne de caractères formée de deux parties : le jour **J** et le mois **M** séparés par le caractère "/" (On ne tiendra compte que des contrôles suivants : **J** est un entier compris entre 1 et 31 et **M** est un entier compris entre 1 et 12).
3. Déterminer dans un tableau **CODE** les **N** codes à partir du tableau **DATE** en utilisant le principe ci-après, sachant qu'un code est une chaîne de **4** chiffres :
   - Concaténer **J** et **M** pour former un nombre de **2**, **3** ou **4** chiffres.
   - Multiplier ce nombre par un entier aléatoire **X** compris entre **5** et **64** afin d'obtenir un nouveau nombre **Y**.
   - Former le code de quatre chiffres comme suit :
     - Si **Y** est un entier de **4** chiffres, déplacer le chiffre des milliers vers la droite de ce nombre.
     - Si **Y** est inférieur à 1000, ajouter des zéros (**0**) à droite de ce nombre pour que sa longueur soit égale à **4**.
     - Si **Y** est supérieur à 9999, additionner le nombre formé des 4 chiffres de gauche au nombre formé des chiffres restants à droite jusqu'à ce que sa longueur soit égale à **4**.
4. Afficher les informations relatives à chacune des cartes sous la forme suivante :

Identifiant de la carte : Date de sa création correspondante : Code correspondant

NB : Le candidat n'est pas appelé à vérifier l'unicité des identifiants dans le tableau **IDENT**.

Exemple :

Pour **N**=3 et les deux tableaux **IDENT** et **DATE** suivants :

IDENT	12345678	23456789	34567891
	1	**2**	**3**
DATE	8/11	2/1	24/12
	1	**2**	**3**

❖ On obtient le tableau **CODE** suivant :

CODE	1108	6720	1314
	1	**2**	**3**

En effet, les codes des cartes sont obtenus comme suit :

- Pour la carte **N°1**, la concaténation de **J** et de **M** donne le nombre **811**. En supposant que l'entier aléatoire **X = 10**, le nombre obtenu est égal à **8110** (811 * 10) qui est composé de 4 chiffres. En déplaçant le chiffre des milliers (**8**) vers la droite, on obtient le code **1108**.
- Pour la carte **N°2**, la concaténation de **J** et de **M** donne le nombre **21**. En supposant que l'entier aléatoire **X = 32**, le nombre obtenu est égal à **672** (21 * 32) qui est inférieur à 1000. En ajoutant un zéro à droite pour que sa longueur soit égale à **4**, on obtient le code **6720**.
- Pour la carte **N°3**, la concaténation de **J** et de **M** donne le nombre **2412**. En supposant que l'entier aléatoire **X = 53**, le nombre obtenu est égal à **127836** (2412 * 53) qui est supérieur à 9999. Le nombre formé des 4 chiffres de gauche (**1278**) sera additionné au nombre formé des chiffres restants à droite (**36**), on obtient le code **1314** (1278 + 36).

❖ Le programme affiche :

12345678 : 8/11 : 1108

23456789 : 2/1 : 6720

34567891 : 24/12 : 1314

Travail demandé :

1. Analyser le problème en le décomposant en modules.
2. Ecrire les algorithmes et les tableaux de déclaration des objets relatifs aux modules envisagés.

RÉPUBLIQUE TUNISIENNE **MINISTÈRE DE L'ÉDUCATION** **EXAMEN DU BACCALAURÉAT**	**Session 2019**	
	Épreuve : **INFORMATIQUE**	**Sections : Mathématiques, Sciences expérimentales et Sciences techniques**
	Durée : **1h 30**	Coefficient de l'épreuve : **0.5**

Recommandations, explications, corrigé et barème de notation

Exercice 1 (*4 points = 0.25 × 16*)

(Recommandations et explications)

1)

a)

- ➔ Il est fortement recommandé de lire rapidement l'algorithme proposé et de voir ce qui est demandé de faire dans les questions.
- ➔ Il s'agit de répondre si des propositions de types de variables sont correctes ou fausses.
  il faut à ce moment après avoir lu les questions entrer dans les détails de l'algorithme proposé et remarquer les différentes manipulations sur les variables :

Remarquer que :

La variable C2 est parcourue via des indices dans Majus(C2[i]), ne peut pas être de type tableau sinon on ne peut pas faire lire(C2) et donc le type possible pour cette variable est chaine de caractère.

- ➔ Ce qui élimine la proposition 2

Majus(C1) témoigne que la fonction Majus est appliquée sur une variable de type caractère (C1) → Ce qui élimine les propositions 1 et 3

La variable C3 est initialisée à 0 elle ne peut être qu'entière ou réelle.

La proposition 3 est correcte d'après les remarques citées supra.

b)

Remarquer d'après les propositions qu'on s'intéresse à la variable C3, la valeur de cette variable est déterminée par l'expression C3 ← C3 + 1. Cette expression est exécuté dans la boucle chaque fois qu'on trouve le caractère donné par la variable C1 dans une position dans la variable C2 (Majus(C2[i]) = Majus(C1))→ ce qui permet de compter alors le nombre de fois ou le caractère défini par C1 apparait dans C2.

On n'a pas le problème de majuscule/minuscule puisque tout est mis en majuscule. → La deuxième proposition est vraie

Les autres propositions ne sont pas correctes :

Le nombre de chiffres… : aucune indication sur les chiffres

Caractères en communs… : C1 présente un seul caractère

Le nombre de caractères majuscules de… : Ne peut être vraie puisque Majus() transforme en majuscule déjà après quoi on effectue le compte.

2)

a)

Remarquer que la suite est définie par son premier terme U0 et le Terme général Un. Ce dernier est défini par le terme qui le précède Un-1.

1)

```
T[1] ← 1
Pour i de 2 à n+1  faire
    T[i] ← 1+1/T[i-1]
Fin pour
Un ← T[n+1]
```

→ remarquer qu'on a mis le premier élément U0 de la suite dans la première case du tableau T[1], l'élément U1 sera mis dans la case T[2]…l'élément Un sera mis dans la case T[n+1]

Donc cette suite d'instruction donne bien le résultat et calcule le terme Un

2)

```
U0 ← 1
Pour i de 1 à n Faire
   Un ← 1+1/U0
Fin pour
```

➔ Cette suite d'instruction ne peut pas donner le résultat puisque le terme (Un) sera toujours égal à 2 du début jusqu'à la fin..

3)

```
Un ← 1
Pour i de 1 à n Faire
   Un ← 1+1/Un
Fin pour
```

➔ Cette suite d'instruction donne bien le résultat puisque le terme (Un) change à chaque fois et le nouveau terme est bien donné en fonction du terme précédent. Exemple : pour i=1 ; la première exécution de la boucle donne Un = 1+1/1 = 2 ; la deuxième exécution donne Un=1+1/2 …etc. Ce qui donne bien le résultat.

4)

```
U0 ← 1
Pour i de 1 à n Faire
   Un ← 1+1/U0
   U0 ← Un
Fin pour
Un ← U0
```

➔ Cette suite d'instruction donne bien le résultat puisque à chaque exécution de la boucle le terme U0 prend la valeur du terme précédent et (Un) est calculé en fonction du terme précédent.

b)

La seule entête qui donne le résultat est bien DEF FN Suite (n : Entier) : Réel

Puisque n est entier et le résultat de calcul du nième terme de la suite est bien réel.

CORRIGÉ

Exercice 1 (*4 points*)

1)

a)

Objet	Type/nature
C1	Chaîne
C2	Chaîne
C3	Entier

F

Objet	Type/nature
C1	Caractère
C2	Caractère
C3	Entier

F

Objet	Type/nature
C1	Chaîne
C2	Caractère
C3	Réel

F

Objet	Type/nature
C1	Caractère
C2	Chaîne
C3	Entier

V

b)

F Ecrire ("Le nombre de caractères majuscules de ",C1," et ",C2," est : ",C3)

V Ecrire ("Le nombre d'occurrences de ",C1," dans ",C2," est : ",C3)

F Ecrire ("Le nombre de chiffres dans ",C2," est : ",C3)

F Ecrire ("Le nombre de caractères communs entre ",C1," et ",C2," est : ",C3)

2)

a) La séquence algorithmique qui permet de déterminer le terme **Un** avec **n ≥ 0** est :

```
T[1] ← 1
Pour i de 2 à n+1  faire
   T[i] ← 1+1/T[i-1]
Fin pour
Un ← T[n+1]
```

V

```
U0 ← 1
Pour i de 1 à n Faire
   Un ← 1+1/U0
Fin pour
```

F

```
Un ← 1
Pour i de 1 à n Faire
   Un ← 1+1/Un
Fin pour
```

V

```
U0 ← 1
Pour i de 1 à n Faire
   Un ← 1+1/U0
   U0 ← Un
Fin pour
Un ← U0
```

V

b) L'entête de la fonction qui permet de déterminer le terme **Un** avec **n ≥ 0** est :

DEF FN Suite (n : Réel) : Réel **F**

DEF FN Suite (n : Entier) : Réel **V**

DEF FN Suite (n : Entier) : Entier **F**

DEF FN Suite (n : Réel) : Entier **F**

Exercice 2 (*3 points = 0.5 ×3 + 0.5 + 0,5×2*)
(Recommandations et explications)

1)

x	nb
5403	**4**
176	**3**
3	**1**

2)
Chaque fois que le quotient de la division de x par 10 est différent de zéro on ajoute 1 à nb et on aura un nouveau quotient x ← x Div 10 cela se termine lorsque ce dernier (le quotient)sera égal à zéro
Exemple 8 div 10 = 0
Donc cela calcule bien le nombre de chiffres(ou digit) qui composent un nombre.

3)
On convertit le nombre en une chaine et puis on compte le nombre de caractères qui sera la longueur de la chaine..

CORRIGÉ

1)

x	nb
5403	**4**
176	**3**
3	**1**

2) Cette séquence permet de déterminer **le nombre de chiffres** d'un **entier x** donné.

3) **Convch(x, xch)**
nb ← Long(xch)
NB. -0,25 par erreur

Problème :
(Recommandations et explications)

Lire attentivement l'énoncé du problème et dégager le résultat final demandé →
Il s'agit d'afficher les informations relatives à chacune des cartes comme c'est demandé dans l'étape 4

On pourra créer un module pour l'affichage des : Identifiant de la carte, de la Date de sa création correspondante et le Code correspondant. Puisque d'après l'énoncé toutes ces informations sont à extraire à partir des tableaux. Appelons ce module **Afficher**

Remarque : Il faut voir le problème dans son globalité et remarquer qu'il s'agit bien en fin de compte de trois tableaux : Un pour stocker les identifiants, un pour stocker les dates et un pour stocker les codes.

Le parcours de ces tableaux en même temps donne le résultat voulu : Pour un indice i qui va de 1 à n on parcourt simultanément les trois tableaux.

Si on cherche dans l'énoncé on trouve bien les façons pour remplir ces 3 tableaux

Le tableau des identifiants c'est à remplir comme indiqué dans l'étape 1 dans l'énoncé

Le tableau des dates c'est à remplir comme indiqué dans l'étape 2 de l'énoncé

Le tableau des codes c'est à remplir comme indiqué dans l'tape 3 de l'énoncé

Pour les deux premiers tableaux on pourra se servir d'un seul module pour le remplissage. Appelons le **Remplir**

Pour le troisième tableau on pourra se servir d'un module pour le remplir. Appelons le **Coder**

Remarques :
- Des contrôles de saisie sont à prévoir dans le module Remplir pour le remplissage des tableaux des identifiants et le tableau des dates.
(On pourra aussi créer des modules de remplissage pour chaque tableau mais cette solution est à éviter pour gagner du temps, un seul module est suffisant)
- Il est à dire que les contrôles de saisie empêchent les utilisateurs des programmes de saisir des données en dehors de ce qui est permis ou des valeurs permises comme par exemple saisir le mois numéro 15 qui n'existe pas bien sûr.
- Remarquer aussi que dans le cadre des contrôles de saisie que dans la première étape de l'énoncé il y a la phrase : **sachant que l'identifiant est une chaîne formée de 8 chiffres** donc on doit vérifier après chaque saisie si la chaine saisie par l'utilisateur est formée seulement par des chiffres. On pourra prévoir un module pour la vérification si le numéro saisi contient seulement des chiffres, appelons ce module Verifnum, ce dernier va être utilisé dans le module Remplir.

Maintenant qu'on a compris de quoi il s'agit il est simple de passer à l'élaboration des différents modules.

Commençons par le programme principal puis élaborons les différents modules :

CORRIGÉ

1) Analyse du programme principal :

Nom : Validation
Résultat = PROC Afficher (IDENT, DATE, CODE, n)
(IDENT, DATE, CODE, n) = PROC Remplir (IDENT, DATE,n)
PROC Coder (CODE, DATE, n)
Fin Validation

TDNT

Type
tab= tableau de 50 chaînes

TDOG

Objet	Type/Nature	Rôle
IDENT	tab	Contient les identifiants des cartes
DATE	tab	Contient les dates de création des cartes
CODE	tab	Contient les codes des cartes
n	Entier	Le nombre de cartes
Remplir	Procédure	Saisir n et remplir les tableaux Ident et Date
Coder	Procédure	Déterminer les codes des différentes cartes
Afficher	Procédure	Afficher les éléments des trois tableaux

2) Les algorithmes des modules :

```
0) DEF PROC Remplir (var Id,D:tab ; var n:entier)
1) Répéter
       Ecrire ("Donner le nombre de cartes : ")
       Lire(n)
   Jusqu'à (n dans [3..50])
2) Pour i de 1 à n faire
       Répéter
           Ecrire ("Donner l'identifiant de la carte n° ",i, " : ")
           Lire(Id[i])
       Jusqu'à (Long(Id[i]=8)) et (FN Verifnum(Id[i]))
       Ecrire ("Donner la date en jour et en mois de la carte n° ",i, " : ")
       Répéter
           Ecrire ("Donner J : ")
           Lire(j)
       Jusqu'à ( j dans [1..31])
       Répéter
        Ecrire ("Donner M : ")
        Lire(m)
       Jusqu'à  (m dans [1..12])
       Convch(j,jch)
       Convch(m,mch)
       D[i] ← jch+"/"+mch
   Fin pour
3) Fin Remplir
```

TDOL

Objet	Type/Nature	Rôle
i	Entier	Compteur
m	Entier	Contient le mois
j	Entier	Contient le jour
jch	chaîne	Résultat de la conversion du jour en chaîne
mch	chaîne	Résultat de la conversion du mois en chaîne
Verifnum	Fonction	Vérifier si une chaîne est numérique

DEF FN Verifnum (ch:chaine) : booléen

```
0) k←0
   Répéter
      k ←k+1
   Jusqu'à (non ch[k] dans ["0". ."9"]) ou (k=Long(ch))
1) Verifnum ← ch[k] dans ["0". ."9"]
2) Fin Verifnum
```

TDOL

Objet	Type/Nature	Rôle
k	Entier	Compteur

```
0) DEF PROC Coder (var C:tab ; D:tab ; n:entier)
1) Pour i de 1 à n faire
        ch ←D[i]
        Efface (ch,Pos("/",ch),1)
        Valeur (ch,r,e)
        x←5+ Aléa(60)
        nb← r*x
        Convch(nb,nbch)
        Si (Long (nbch)=4)
           Alors C[i] ← Sous_chaine(nbch,2,4) + nbch[1]
              Sinon
                 Si (Long(nbch) < 4)
                    Alors
                    Répéter
                       nbch ← nbch +"0"
                    Jusqu'à (Long(nbch) = 4)
                    C[i]← nbch
                 Sinon
                    Valeur(Sous_chaine(nbch,1,4),nb1)
                    Valeur(Sous_chaine(nbch,5,Long(nbch)),nb2)
                    Convch(nb1+nb2, C[i])
                 Fin si
        Fin si
   Fin pour
2) Fin Coder
```

TDOL

Objet	Type/Nature	Rôle
i	Entier	Compteur
r	Entier	Contient les chiffres relatifs à la date
x	Entier	Entier aléatoire
e	Entier	Position de l'erreur
nb	Entier long	Résultat de la multiplication de r par x
nb1	Entier	Les quatre chiffres de gauche de nb
nb2	Entier	Le reste des chiffres de nb
ch	Chaîne	Une date du tableau D
nbch	Chaîne	Résultat de la conversion de nb en chaîne

```
0) DEF PROC Afficher (Id,D,C:tab ; n:entier)
1) Pour i de 1 à n faire
     Ecrire (Id[i], " : ", D[i], " : ", C[i])
   Fin pour
2) Fin Afficher
```

TDOL

Objet	Type/Nature	Rôle
i	Entier	Compteur

Barème détaillé :

Programme principal : - Modularité - Cohérence (appels + conformité des paramètres)	***1.5 points =*** 0.5 1 = (0.5+0.5)
Saisie du nombre de cartes N : - Lecture - Contrôle de la contrainte	***0.5 point =*** 0.25 0.25
Remplissage du tableau IDENT : - Parcours du tableau - Lecture de IDENT[i] - Contrôle des contraintes : 8 chiffres (chiffres + longueur)	***1.75 points =*** 0.25 0.25 1.25 = (1+0.25)
Remplissage du tableau DATE : - Parcours du tableau - Lecture de J + contrôle des contraintes - Lecture de M + contrôle des contraintes - Remplissage d'un élément date (conversion + concaténation + affectation)	***2 points =*** 0.25 0.25+0.25 0.25+0.25 0.75 = (0.25*3)
Formation du tableau CODE : - Parcours du tableau - Concaténation de J et M - Lecture aléatoire de X - Détermination du nombre Y - Formation du code (cas 1 + cas 2 + cas 3) - Remplissage d'un élément Code	***4.25 points =*** 0.25 0.5 0.5 0.25 2.5 = (0.5+0.75+1.25) 0.25
Affichage des éléments des tableaux : - Parcours des tableaux - Affichage des éléments + respect de la forme	***1.25 points =*** 0.25 1 = (0.25*3 + 0.25)
TDNT + TDOG	***0.75 point =*** (0.25+0.5)
TDOL	***1 point***

-0,25 par erreur.

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ●●●●● **EXAMEN DU BACCALAURÉAT**	***Session* 2018**	
	Épreuve : **INFORMATIQUE**	*Sections :* **Mathématiques, Sciences expérimentales, Sciences techniques**
	Durée : **1h 30**	Coefficient de l'épreuve : **0.5**

Section : ..... N° d'inscription : ..... Série : .....

Nom et prénom : .....

Date et lieu de naissance : .....

Signatures des surveillants

.....

.....

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.

Les réponses aux exercices 1 et 2 doivent être rédigées sur cette même feuille qui doit être remise à la fin de l'épreuve

Exercice 1 (4 points)

On donne les informations suivantes :

- **Ch1**, **Ch2** et **Ch** de type chaine,
- **P** de type réel,
- **R** et **Q** de type entier.
- **Ch1** est initialisée à "Bac", **Ch2** à "2017/2018" et **P** à 19,58.

Compléter la colonne **"Instruction à exécuter"** du tableau ci-dessous, par les instructions algorithmiques permettant d'aboutir aux valeurs finales mentionnées dans le tableau. Les instructions à fournir devront utiliser les variables précédemment données et tenir compte de leur initialisation.

Instruction à exécuter	Valeur finale de la variable
	Ch contient "Bac 2017/2018"
	Q contient 19
	R contient 2017
	Ch contient "Bac 2018"

NE RIEN ECRIRE ICI

Exercice 2 (4 points)

Soient les algorithmes suivants relatifs à des fonctions ayant pour paramètres deux entiers **a** et **b**, strictement positifs, donnés au niveau du programme principal :

```
0) DEF FN F1 (a,b : entier) : entier
1) R←0
2)  Pour i de 1 à b Faire
      R←R+a
    FinPour
3) F1← R
4) Fin F1
```

```
0) DEF FN F2 (a,b : entier) : entier
1) R←1
2)  Pour i de 1 à b Faire
      R←R*a
    FinPour
3) F2← R
4) Fin F2
```

```
0) DEF FN F3 (a,b : entier) : entier
1) P←0
2) Pour i de a à b Faire
      P←P + i
   FinPour
3) F3←P
4) Fin F3
```

```
0) DEF FN F4 (a,b : entier) : entier
1)  TantQue (a≠b) Faire
        Si (a > b) Alors
           a←a-b
        Sinon
           b←b-a
        FinSi
    FinTantQue
2) F4←a
3) Fin F4
```

```
0) DEF FN F5 (a,b : entier) : entier
1) TantQue (a*b≠0) Faire
    Si (a>b) Alors
          a←a MOD b
    Sinon
          b←b MOD a
    FinSi
  FinTantQue
2)  Si (a= 0) Alors
        F5←b
    Sinon
        F5←a
    FinSi
3) Fin F5
```

NE RIEN ECRIRE ICI

Valider chacune des propositions suivantes en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** si elle est fausse.

a. Pour calculer le **produit** de **a** par **b**, on peut faire appel à la (aux) fonction(s) :

☐ F1 ☐ F2 ☐ F3 ☐ F4 ☐ F5

b. Pour calculer $\mathbf{a}^{\mathbf{b}}$, on peut faire appel à la (aux) fonction(s) :

☐ F1 ☐ F2 ☐ F3 ☐ F4 ☐ F5

c. Pour calculer le **PGCD** des deux entiers **a** et **b**, on peut faire appel à la (aux) fonction(s) :

☐ F1 ☐ F2 ☐ F3 ☐ F4 ☐ F5

d. Pour calculer la somme des entiers de l'intervalle **[a..b]**, on peut faire appel à la (aux) fonction(s) :

☐ F1 ☐ F2 ☐ F3 ☐ F4 ☐ F5

Voir suite au verso ☞

Problème (12 points)

Dans le but de gérer les discussions dans un groupe fermé créé au sein d'un réseau social, on se propose de réaliser une application informatique permettant d'afficher à l'administrateur du groupe :

- La liste des adhérents qui ont envoyé un même commentaire donné.
- La liste des adhérents inactifs dans le groupe, sachant qu'un adhérent est qualifié inactif s'il n'a envoyé aucun commentaire.

Pour ce faire, on se propose de :

- Remplir un tableau **A** par les noms d'utilisateurs des **Na** adhérents du groupe avec 5<**Na** ≤50, sachant qu'un nom d'utilisateur est unique et formé par une suite d'au minimum trois caractères commençant obligatoirement par une lettre majuscule.
- Remplir un tableau **M** par les **Nm** commentaires émis par les adhérents de ce groupe avec **2<Nm≤100,** en respectant le format général suivant : **NumE#commentaire**, où :
  - **NumE** est une donnée représentant l'indice de l'émetteur du commentaire dans le tableau **A**
  - **commentaire** représente le texte envoyé par cet émetteur.
- Afficher :
  - le nom d'utilisateur de chaque adhérent qui a envoyé un commentaire donné,
  - la liste des adhérents inactifs.

Exemple : Pour **Na**=5, **Nm**=7 et les tableaux **A** et **M** suivants :

A	Olfa	Hichem_2018	Mohamed	Amir@	Salah 20
	1	2	3	4	5

M	3#Help me	4#Help	1#Mes salutations	3#cours info	3#Algo	1#OK	3#Help
	1	2	3	4	5	6	7

Pour cet exemple M[3] signifie :

1#Mes salutations

Indice (dans le tableau A) de l'émetteur du commentaire

Texte du commentaire

Si le commentaire saisi est "**Help**", alors le programme devra afficher les deux résultats suivants :

Le commentaire **Help** est envoyé par : **Amir@, Mohamed**

La liste des adhérents inactifs est : **Hichem_2018**

Salah 20

Travail demandé :

1) Analyser le problème en le décomposant en modules.
2) Ecrire les algorithmes et les tableaux de déclaration relatifs aux modules envisagés.

CORRIGE DE L'EPREUVE THEORIQUE DE LA MATIERE INFORMATIQUE POUR LES SECTIONS SCIENTIFIQUES

Session 2018

RESUME

Ceci est une proposition d'un corrigé avec des commentaires de l'épreuve d'informatique de la session 2018 concernant les sections scientifiques : Sciences expérimentales, Maths et Sciences techniques.

Corrigé

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ***** EXAMEN DU BACCALAURÉAT	Épreuve : **INFORMATIQUE**	
	Sections : Mathématiques, Sciences Expérimentales et Sciences Techniques	
	Durée : **1 H 30**	Coefficient : **0.5**
Corrigé		

Exercice 1 (4 points= 1*4)

Instruction à exécuter	Valeur finale de la variable
Ch← Concat(Ch1, " ", Ch2)	**Ch** contient "Bac 2017/2018"
Q ← Tronc (P)	**Q** contient 19
Valeur (sous-chaine(Ch2,1,4),R,e)	**R** contient 2017
Effacer (Ch,5,5)	**Ch** contient "Bac 2018"

NB :

- **On acceptera toute autre solution correcte.**
- **-0.25 par erreur (de syntaxe, ordre de paramètres, confusion Pascal/Algorithmique, etc.)**

Exercice 2 (4 points = (0.2*5)*4)

Pour cet exercice, seules les réponses V, F, Vrai, Faux ont été acceptées.

Valider chacune des propositions suivantes en mettant dans la case correspondante la lettre **V** si elle est correcte ou la lettre **F** si elle est fausse.

a. Pour calculer le **produit** de **a** par **b**, on peut faire appel à la (aux) fonction(s) :

F1 [V] F2 [F] F3 [F] F4 [F] F5 [F]

b. Pour calculer $\mathbf{a^b}$, on peut faire appel à la (aux) fonction(s) :

F1 [F] F2 [V] F3 [F] F4 [F] F5 [F]

c. Pour calculer le **PGCD** des deux entiers a et b, on peut faire appel à la (aux) fonction(s) :

F1 [F] F2 [F] F3 [F] F4 [V] F5 [V]

d. Pour calculer la somme des entiers de l'intervalle [a..**b**], on peut faire appel à la (aux) fonction(s) :

F1 [F] F2 [F] F3 [V] F4 [F] F5 [F]

Problème :

1. **Analyse du programme principal :**

   Nom : Groupe

   Resultat = *PROC Recherche(M,A, na,nm)*

   PROC Inactifs (Occ, A, na)

   M, nm, Occ= PROC RemplirM (M,Occ,na,nm)

   A,na=PROC RemplirA(A,na)

   Fin Groupe

TDNT

Tab1= Tableau [1..100] de chaine de caractères
Tab2 = Tableau [1..100] d'entiers

TDOG

Objet	*Type/Nature*	*Rôle*
M	*Tab1*	*Contenant les messages*
A	*Tab1*	*Contenant les noms des adhérents*
Occ	*Tab2*	*Contenant le nombre de messages par adhérent*
nm	*Entier*	*Le nombre de messages*
na	*Entier*	*Le nombre d'adhérents du groupe*
Inactifs	*Procédure*	*Permet d'afficher les membres inactifs*
Recherche	*Procédure*	*Permet d'afficher l'adhérent qui a envoyé un message donné*
RemplirM	*Procédure*	*remplir les messages et déterminer le nombre de messages par adhérent.*
RemplirA	*Procédure*	*Permet de saisir les noms des membres*

2) Les algorithmes

a. Algorithme de la procédure RemplirA

0) DEF PROC RemplirA (var A:Tab1 ; var na : entier)

1) *Répéter*

Ecrire ("donner le nombre d'adhérents du groupe :")

Lire(na)

Jusqu'à (na dans [5..50])

2) Pour i de 1 à na faire

Répéter

Ecrire("Donner le nom de l'adhérant numéro", i, " : ")

Lire(A[i])

Jusqu'a ((A[i][1] dans ["A".. "Z"]) ET (long(A[i]) ≥3) ET((i>1) ET NON(FN Existe(A[i],A,i-1))

FinPour

4) Fin RemplirA

TDOL

Objet	***Type/Nature***	***Rôle***
i	*entier*	*compteur*
Existe	*fonction*	*Vérifier l'existence d'un adhérant dans A*

*0)**DEF FN Existe(ch: chaine ; t:Tab1; indice: entier):boolean***
1) i←0
2) Répéter
i ← i+1
Jusqu'à ((t[i]=ch) ou(i=indice)
3) Existe ← (t[i]=ch)
4) Fin Existe

TDOL

Objet	Type/Nature	Rôle
i	entier	compteur

b. Algorithme de la procédure RemplirM

*0) **DEF PROC RemplirM (A : Tab1 ;var M:Tab1 ; var Occ:Tab2; na:entier; var nm:entier)***
1) Répéter
Ecrire ("donner le nombre de commentaires : ")
Lire (nm)
Jusqu'à (nm dans [3..100])
2) Pour i de 1 à na faire
Occ[i] ←0
Fin pour
3) Pour i de 1 à Nm faire
Ecrire ("donner le commentaire")
Lire (message)
Répéter
Ecrire ("donner le numéro de l'émetteur")
Lire (NumE)
Jusqu'à (NumE dans [1..Na])
Convch(NumE,ch)
M[i] ← Concat (ch ,"#",message)
Occ[NumE] ← Occ[NumE] +1
Fin pour
*4) **Fin RemplirM***

TDOL

Objet	*Type/Nature*	*Rôle*
NumE	*entier*	*Numéro de l'émetteur*
i	*entier*	*compteur*
message	*Chaine de caractères*	*Message à envoyer*

c. Algorithme de la procédure Inactifs

*0) **DEF PROC Inactifs (Occ:Tab2; A:tab1 ; na:entier)***
1) Ecrire ("la liste des adhérents inactifs est : ")
2) Pour i de 1 à na faire
Si(Occ[i]= 0) Alors
Ecrire (A[i])
FinSi
Fin Pour
*3) **Fin Inactifs***

TDOL

Objet	*Type/Nature*	*Rôle*
i	*entier*	*compteur*

d. Algorithme de la procédure Recherche

0) ***DEF PROC Recherche (M,A:Tab1; na,nm :entier)***
1) Ecrire ("Donner le commentaire à chercher :"), lire(msg)
2) Pour i de 1 à nm faire
Si (Sous-chaine(M[i], pos("#",M[i])+1, long(M[i])- pos("#",M[i]))= msg) alors
Valeur(Sous-chaine(M[i],1, pos("#",M[i])-1) ,Nume,e)
Ecrire(A[Nume])
Fin si
Fin pour
3) ***Fin PROC Recherche***

TDOL

Objet	*Type/Nature*	*Rôle*
i	*entier*	*compteur*
Nume	*entier*	*Numéro émetteur*
e	*entier*	*Position de l'erreur*
msg	*chaine*	*Message à chercher*

NB :

- ***Il est à noter que ceci n'est qu'une proposition de solution.***
- ***Il existe d'autres solutions possibles pour résoudre ce problème et qui ont été acceptées.***

Barème détaillé du problème :

P. P : - ***Modularité*** - ***Cohérence (appels + conformité des paramètres))***	***1.5 points =*** ***0.5*** ***1=(0.5 appels +0.5 conformité)***
Le remplissage du tableau A avec respect des contraintes : - ***La saisie du nombre d'adhérents avec respect de la contrainte*** - ***Parcours du tableau*** - ***Lecture de A[i] + Respect des contraintes (1er caractère, longueur, unicité)***	***1.75 points =*** ***0.5= 0.25 saisie+ 0.25 contrainte*** ***0.25*** ***1 = 0.25 + 0.25*3***
Traitement de l'unicité	***1.25 point (-0.25 par erreur)***
Le remplissage du tableau M avec respect des contraintes : - ***La saisie du nombre de commentaires avec respect des contraintes*** - ***Parcours du tableau*** - ***Lecture de M[i] : Test du numéro de l'émetteur+ # +lecture du commentaire+ affectation***	***2 points =*** ***0.5 = 0.25 + 0.25*** ***0.25*** ***1.25 = 0.5 +0.25+ 0.25+0.25***
Affichage : - ***Détermination des adhérents inactifs (2 parcours+ test+ affichage)*** - ***Saisie du commentaire*** - ***Recherche des adhérents qui ont envoyé le commentaire donné (parcours+ test + extraction de l'indice + affichage)***	***4 points =*** ***1.75 =0.5*2+0.5+0.25*** ***0.25*** ***2=0.25+1+0.5+0.25***
TDNT + TDOG	***0.75 points = 0.25 + 0.5***
TDOL	***0.75 points***

NB : -0.25 par type erreur.

RÉPUBLIQUE TUNISIENNE MINISTÈRE DE L'ÉDUCATION ●●○●● EXAMEN DU BACCALAURÉAT	Épreuve : **INFORMATIQUE** Sections : **Mathématiques, Sciences expérimentales, Sciences Techniques** Durée : **1 H 30** — Coefficient : **0.5** **SESSION 2017**

Section : ..... N° d'inscription : ..... Série : .....

Nom et prénom : .....

Date et lieu de naissance : .....

Signatures des surveillants

.....

.....

Le sujet comporte 3 pages numérotées de 1/3 à 3/3.

La réponse à l'exercice 1 doit être rédigée sur cette même feuille qui doit être remise à la fin de l'épreuve

Exercice N°1 : (2.5 points)

Réécrire dans la colonne « **Correction** », les phrases données dans la colonne « **Proposition** » en apportant les modifications nécessaires afin qu'elles soient correctes :

Proposition	Correction
La recherche dichotomique est une méthode de recherche appliquée sur un tableau quelconque.	
Un tableau est une structure de données permettant de regrouper plusieurs éléments de types différents.	
Les paramètres formels sont utilisés lors de l'appel d'un sous-programme.	
L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de gauche à droite ou de droite à gauche.	
Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs appartenant à un type prédéfini.	

Exercice 2: (5.5 points)

Soit l'algorithme "**Exercice**" suivant :

```
0) Début Exercice
1) Lire(ch1)
   Lire(ch2)
   Répéter
      Lire (p)
   Jusqu'à (p ≤ long(ch1))
2) ch←""
   Pour i de 1 à p-1 faire
      ch←ch+ch1[i]
   Fin pour
3) ch← ch+ ch2
4) Pour i de p à long (ch1) faire
      ch←ch+ ch1[i]
   Fin pour
5) ch1←ch
6) Ecrire(ch1)
7) Fin Exercice
```

a) Dresser le tableau de déclaration des objets de cet algorithme.

b) Proposer pour chacun des blocs d'instructions n°**2** et n°**4**, une expression équivalente utilisant un ou plusieurs modules prédéfinis : procédure(s) et / ou fonction(s) prédéfinie(s).

c) Quel sera le résultat affiché par cet algorithme pour **ch1 = "BAC 2017"**, **ch2 = "2016/"** et **p = 5** ?

d) Déduire le rôle de cet algorithme.

e) Quel est le module prédéfini qui peut jouer le même rôle que cet algorithme ? Proposer l'appel correspondant.

Problème : (12 Points)

Une station de radio lance chaque début de semaine un concours hebdomadaire intitulé « Hitparade » pour classer cinq titres de chansons qu'elle propose à ses auditeurs et affiche le résultat du classement le samedi.

Tout au long de la semaine, un responsable de la station reçoit par téléphone les propositions des participants au concours parmi ses auditeurs qui sont appelés à choisir le titre préféré afin d'ajouter à son score 3 points.

Pour obtenir le classement final des cinq chansons suite à la participation d'un nombre donné des auditeurs de la station, on se propose d'écrire un programme qui permet de :

- Remplir un tableau **T1** par les cinq titres des chansons, sachant qu'un titre est composé uniquement par des lettres alphabétiques et des espaces.
- Saisir le nombre **N** de participants avec $5 \leq N \leq 100$.
- Générer un tableau **T2** représentant les scores des cinq chansons en ajoutant 3 points au score de chacune si le numéro qui lui correspond a été choisi par un participant. Ce numéro est un chiffre allant de 1 à 5, représentant l'emplacement de la chanson dans le tableau **T1**.
- Afficher le classement des chansons, comme indiqué dans l'exemple ci-après, en commençant par le titre de la chanson ayant le plus grand score. Il est à noter que les chansons ayant un même score auront un même rang dans le classement.

Exemple :

Pour :

T1	Happy	Sorry	Me Quemo	Rosa	Hello
	1	2	3	4	5

Et un nombre de participants **N = 11** ayant fait les choix suivants :

Choix du participant n°1 : 5
Choix du participant n°2 : 4
Choix du participant n°3 : 5
Choix du participant n°4 : 1
Choix du participant n°5 : 2
Choix du participant n°6 : 5
Choix du participant n°7 : 1
Choix du participant n°8 : 2
Choix du participant n°9 : 4
Choix du participant n°10 : 1
Choix du participant n°11 : 5

On aura :

T2	9	6	0	6	12
	1	2	3	4	5

Le programme affiche :

Le classement est :

Rang 1 : Hello

Rang 2 : Happy

Rang 3 :Sorry, Rosa

Rang 4 : Me Quemo

Questions :

1) Analyser le problème en le décomposant en modules.
2) Ecrire les algorithmes et les tableaux de déclaration relatifs aux modules envisagés.

Corrigé Informatique Sections : Mathématiques, Sciences expérimentales, Sciences Techniques

Session principale Baccalauréat 2017

Exercice N°1 :

Réécrire dans la colonne « **Correction** », les phrases données dans la colonne « **Proposition** » en apportant les modifications nécessaires afin qu'elles soient correctes :

Proposition	Correction
La recherche dichotomique est une méthode de recherche appliquée sur un tableau quelconque.	• La recherche dichotomique est une méthode de recherche appliquée sur un tableau **trié**. • La recherche **séquentielle** est une méthode de recherche appliquée sur un tableau quelconque.
Un tableau est une structure de données permettant de regrouper plusieurs éléments de types différents.	Un tableau est une structure de données permettant de regrouper plusieurs éléments de **même** type.
Les paramètres formels sont utilisés lors de l'appel d'un sous-programme.	• Les paramètres formels sont utilisés lors de **la définition** d'un sous-programme **(au niveau de l'entête)**. • Les paramètres **effectifs** sont utilisés lors de l'appel d'un sous programme.
L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de gauche à droite ou de droite à gauche.	L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de **gauche à droite**.
Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs appartenant à un type prédéfini.	• Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs **n'appartenant pas à un type prédéfini dites identificateurs**. • Le **type intervalle** définit un ensemble ordonné et fini de valeurs appartenant à un type prédéfini.

Exercice 2 :

a) Tableau de déclaration des objets de l'algorithme « Exercice »

T.D.O

Objet	Type/nature	Rôle
ch1	Chaîne	La première chaîne à saisir
ch2	Chaîne	La deuxième chaîne à saisir
p	Entier	Position
ch	Chaîne	Chaîne intermédiaire
i	Entier	Compteur

b) expression équivalente utilisant un ou plusieurs modules prédéfinis

Séquence n°2 : ***ch ← Sous-chaine (ch1, 1, p-1)***

Séquence n°4 : ***ch ← Concat (ch, Sous-chaine (ch1,p, Long(ch1)-p+1))***

c) Le programme affiche : BAC 2016/2017

d) Cet algorithme permet d'insérer une chaine ch2 dans une chaine ch1 à une position p.

e) le module prédéfini qui peut jouer le même rôle est : Insère (ch2, ch1, p)

Problème :

1) Analyse du programme principal

Nom : Hitparade
Résultat= PROC Afficher (T1, T2)
(T1,T2)= [PROC Chanson (T1)
PROC Score (T2,N)]
PROC Trier (T1,T2)
N= PROC Saisir(N)
Fin Hitparade

TDNT

Type
Tab1= tableau de 5 chaînes
Tab2= tableau de 5 entiers

TDOG

Nom	***Nature/Type***	***Rôle***
T1	*Tab1*	*Tableau des chansons*
T2	*Tab2*	*Tableau des scores*
N	*Entier*	*Nombre de participants*
Saisir	*Procédure*	*Saisir le nombre des participants*
Chanson	*Procédure*	*Remplir le tableau des chansons*
Score	*Procédure*	*Comptabiliser les scores*
Trier	*Procédure*	*Trier les scores et les chansons*
Afficher	*Procédure*	*Afficher le résultat*

2) Les algorithmes des modules envisagés :

a) Module *Chanson :*

```
0)  DEF PROC Chanson (Var T1 : Tab1)
1)  Pour i de 1 à 5 faire
      Répéter
          Ecrire (" Saisir le titre de la chanson n°", i, " :")
          Lire (T1[i])
      Jusqu'à (FN Verif (T1[i]))
    Fin pour
2)  Fin Chanson
```

TDOL

Nom	Nature/Type	Rôle
i	Entier	Compteur
Verif	Fonction	Vérifier si une chaîne est composée uniquement de lettres et des espaces

b) Module *Verif :*

```
0)  DEF FN Verif (ch : chaine) : booléen
1)  j ← 0
    Répéter
      j ← j + 1
    Jusqu'à (NON (Majus (ch[j]) dans ["A".."Z", "  "])) OU ( j=Long(ch))
2)  Verif ← (Majus (ch[j]) dans ["A".."Z", " "]
3)  Fin Verif
```

TDOL

Nom	Type	Rôle
j	Entier	Compteur

c) Module *Saisir :*

```
0)  DEF PROC Saisir (Var N : entier)
1)  Répéter
            Ecrire (" Saisir le nombre de participants : ")
            Lire (N)
    Jusqu'à (N dans [5..100])
2)  Fin Saisir
```

d) Module *Score :*

```
0)  DEF PROC Score (Var T2 : Tab2 ; N : entier )
1)  Pour i de 1 à 5 faire
        T2[i] ← 0
    Fin pour
2)  Pour i de 1 à N faire
        Répéter
            Ecrire (" Saisir le numéro de la chanson :")
            Lire (num)
        Jusqu'à (num dans [1..5])
        T2[num] ← T2[num] + 3
    Fin pour
3)  Fin Score
```

TDOL

Nom	Nature/Type	Rôle
i	Entier	Compteur
num	Entier	Numéro de la chanson choisie

e) Module *Trier :*

```
0)  DEF PROC Trier (Var T2 : Tab2 ; Var T1 : Tab1 )
1)  Pour i de 1 à 4 faire
        Pmax ←FN Posmax (T2, i)
            Si ( T2[Pmax]<>T2[i])
            Alors
              Aux1 ← T2[Pmax]
              T2[Pmax] ← T2[i]
              T2[i] ← Aux1
              Aux2 ← T1[Pmax]
              T1[Pmax] ← T1[i]
              T1[i] ← Aux2
        Fin Si
    Fin pour
2)  Fin Trier
```

TDOL

Nom	***Type***	***Rôle***
i	*Entier*	*Compteur*
Pmax	*Entier*	*Position maximum*
Aux1	*Entier*	*Variable intermédiaire*
Aux2	*Chaine*	*Variable intermédiaire*
Posmax	*Fonction*	*Renvoie la position du maximum*

f) Module *Posmax :*

```
0)  DEF FN Posmax (T2 : Tab2 ; i : entier)
1)  Pm←i
2)  Pour j de i+1 à 5 faire
      Si (T2[Pm] < T2[j])
          Alors Pm←j
      Fin Si
    Fin pour
3)  Posmax ← Pm
4)  Fin Posmax
```

TDOL

Nom	***Nature/Type***	***Rôle***
j	*Entier*	*Compteur*
Pm	*Entier*	*Position maximum*

g) Module *Afficher :*

```
0)  DEF PROC Afficher (T1 : Tab1;T2 : Tab2 )
1)   Ecrire ("le classement est :")
     Rang←1
     Ecrire ("Rang  ",Rang, " :", T1[1])
    Pour i de 2 à 5 faire
      Si (T2[i-1] =T2[i] )
      Alors Ecrire (" ,  ",T1[i])
    Sinon
       Rang ←Rang+1
      Ecrire ("Rang ",Rang, " :",T1[i])
    Fin Si
   Fin pour
2)   Fin Afficher
```

TDOL

Nom	***Nature/Type***	***Rôle***
i	*Entier*	*Compteur*
Rang	*Entier*	*Rang de la chanson*

Examen du baccalauréat session juin 2016

Correction du sujet théorique d'informatique

SECTIONS : Mathématiques + Sciences Expérimentales + Sciences Techniques

Exercice 1 : (3points= 0,25*3*4)

Valider chacune des propositions ci-dessous en mettant dans la case correspondante la lettre (V) si elle est correcte ou la lettre (F) dans le cas contraire.

1) L'identificateur d'une variable :

   [V] ne doit pas commencer par un chiffre.

   [F] peut contenir un espace.

   [V] peut contenir le caractère souligné (tiret bas "_").

2) L'instructionqui permet d'affecter à une variable X, une valeur aléatoire de l'intervalle [2,10] est :

   [F] X ← 2 + Aléa (10)

   [V] X ← 2 + Aléa (9)

   [F] X ← Aléa (2 + 10)

3) Une structure de données tableau peut :

   [F] contenir des éléments de types différents.

   [F] être déclarée avec une taille maximale variable.

   [V] avoir des indices de cases de type caractère.

4) Pour le typescalaire énuméré :

   [F] les valeurs énumérées peuvent appartenir à un type prédéfini.

   [V] une valeur énumérée peutêtre affectée à une variable du même type.

   [V] les opérateurs relationnels "<", ">" et "=" peuvent être appliqués.

Exercice 2 (5 points)

1) A partir des algorithmes donnés ci-dessus, remplir la 2ème colonne du tableau suivant par un exemple de chaqueélément cité dans la 1ère colonne :**(1points = 4 * 0.25)**

Elément	Exemple
Expression booléenne	**(C = "") ou (E ≠ 0)**
	E=0
	C = ""
	E ≠ 0
Procédure prédéfinie	**Valeur(C[1],x,e)**
	Efface(C,1,1)
Paramètre formel	**C**
Paramètre effectif	**A**

Remarque : On accepte aussi **FN inconnue(A)**comme un exemple d'expression booléenne

2) Compléter l'entête de la fonction **Inconnue** par les types appropriés :**(0,5 point= 0.25*2)**

Def Fn **Inconnue** (C : ***Chaîne***) : ***Booléen***

3) Compléter le tableau de déclaration ci-dessous par les types des objets locaux de la fonction **Inconnue. (0,5 point = 0,25*2)**

Objet	*Type / Nature*
X	***Entier / Réel***
E	***Entier***

4) Parmi les variables **A**,**C**, **X** et **E**, réécrire dans le tableau ci-dessous celles qui ne sont pas visibles par le programme principal.**(0.5 point)**

(-0.25 par erreur)

Variables non visibles par le programme principal
C, X, E

5) Donner le résultat affichépar le programme **Exercice** pour chacune des valeurs de **A** suivantes : **(2 points = 4 * 0.5)**

- A = "523" **→523 Vérifie la propriété.**
- A = "-523" **→-523 Ne vérifie pas la propriété.**
- A = "5,23" **→5,23 Ne vérifie pas la propriété.**
- A = "A5B3" **→A5B3 Ne vérifie pas la propriété.**

6) En déduire le rôle de la fonction **Inconnue. (0.5 point)**

La fonction Inconnue permet de **vérifier si une chaîne est formée** uniquement **par des chiffres.**

Problème : (12 points)

1°) Analyse du programme principal

Résultat = PROC Affiche (N,Tx,Ty)
(Tx,Ty)= PROC Tri (N,Tx,Ty,Td)
Td=PROC Distance (N,Xd,Yd,Tx,Ty,Td)
(N,Xd,Yd,Tx,Ty)=PROC Saisie (N,Xd,Yd,Tx,Ty)

Tableau de déclaration des nouveaux types

Type
Tab = tableau de 50 réels

Tableau de déclaration des objets globaux

Objet	**Type / Nature**	**Rôle**
N	Entier	Nombre de bateaux les plus proches
Xd,Yd	Réel	Coordonnées du bateau en détresse
Tx	Tab	Tableau des abscisses des bateaux
Ty	Tab	Tableau des ordonnées des bateaux
Td	Tab	Tableau des distances

Affiche	Procédure	Procédure qui permet de faire l'affichage
Tri	Procédure	Procédure qui permet de faire le tri de Tx, Ty et Td
Distance	Procédure	Procédure qui permet de remplir le tableau des distances Td
Saisie	Procédure	Procédure qui permet de faire la saisie de N, Xd,Yd, Tx et Ty

2°) Analyses des modules envisagés

Analyse de la procédure Saisie

```
DEF PROC Saisie(Var N : entier ; VAR Xd,Yd : réel ; Var Tx,Ty : Tab)
Résultat = (Xd,Yd,N,Tx,Ty)
Xd,Yd = Donnée ("Introduire les coordonnées du bateau en détresse : ")
N=[]Répéter
            N=donnée ("Donner le nombre de bateaux les plus proches : ")
Jusqu'à N dans [1..50]
(Tx ,Ty) = [Répéter
                    Tx[1] ,Ty[1] = Donnée("Introduire les coordonnées du bateau proche n° 1: ")
Jusqu'à ((Tx[1]<>Xd) OU (Ty[1]<>Yd))]
Pour i de 2 à N faire
      Répéter
                    Tx[i] ,Ty[i] = Donnée("Introduire les coordonnées du bateau proche n° ",i," : ")
            Jusqu'à (Non (FNExiste(Tx[i],Ty[i],Tx,Ty,i)) ) ET((Tx[i]<>Xd) OU (Ty[i]<>Yd))
      Fin Pour
Fin Saisie
```

Tableau de déclaration des objets locauxde la procédure Saisie

Objet	Type / Nature	Rôle
i	Entier	Compteur
Existe	Fonction	Fonction qui permet de vérifier l'unicité des coordonnées.

Analyse de la fonction Existe:

```
DEF  FnExiste (x, y: réel ; T1,T2 : tab ; p : entier): booléen
Résultat=   existe ← test
test =[i←0]    Répéter
                    i←i+1
                    test←(T1[i]=x) et  (T2[i]=y)
              Jusqu'à  (i=p-1) ou (test=Vrai)
Fin Existe
```

Tableau de déclaration des objets locaux

Objet	T/N
i	Entier
test	booléen

Analyse de la procédure Distance

```
DEF PROC Distance (N : entier ; Xd,Yd : réel ; Tx,Ty : Tab ; Var Td : Tab)
Résultat = Td
Td= [] Pour i de 1 à N Faire
            Td[i] ←Racine_carré (carrée (Tx[i]-Xd) + carrée (Ty[i]-Yd))
      Fin Pour
Fin Distance
```

Tableau de déclaration des objets locaux de la procédure Distance

Objet	Type / Nature	Rôle
i	Entier	Compteur

Analyse de la procédure Tri

```
DEF PROC Tri (N :entier ; VarTx,Ty,Td : Tab)
Résultat = (Tx,Ty,Td)
(Tx,Ty,Td)=[] Répéter
                    Test ←Faux
                    Pour i de 1 à N-1 Faire
                          Si Td[i] >Td[i+1] Alors
                                        PROC Permut (Td[i],Td[i+1])
                                        PROC Permut (Tx[i],Tx[i+1])
                                        PROC Permut (Ty[i],Ty[i+1])
                                        Test ←Vrai
                          Fin Si
FinPour
                    N← N-1
              Jusqu'à (Test = Faux) OU (N=1)
Fin Tri
```

Tableau de déclaration des objets locaux de la procédure Tri

Objet	Type / Nature	Rôle
i	Entier	Compteur
Test	Booléen	Tester s'il y a eu une permutation ou non
Permut	Procédure	Procédure qui permet de faire la permutation de deux cases.

Analyse de la procédure Permut

```
DEF PROC Permut(Varx,y : réel)
Résultat = (x,y)
(x,y) = aux ← x
        x ← y
        y← aux
Fin Permut
```

Tableau de déclaration des objets locaux de la procédure Permut

Objet	Type / Nature	Rôle
aux	réel	Variable auxiliaire

Analyse de la procédure Affiche

```
DEF PROC Affiche (N : entier ; Tx,Ty : Tab)
Résultat = [ ]  Pour i de 1 à N faire
                    Ecrire ("(",Tx[i]," , " , Ty[i] , ") "    )
              Fin Pour
Fin Affiche
```

Tableau de déclaration des objets de la procédure Affiche

Objet	Type / Nature	Rôle
i	Entier	Compteur

Barème : 12 points

Toute solution équivalente sera acceptée
-0.25 par type d'erreur

Action	Nombre de points
Programme principal : • Modularité • Cohérence (mode de passage, conformité entre nombre, ordre et type des paramètres)	**2 points :** • 1 • 1 = 0.25*4
Saisie de n: • Lecture • Contrôle	**0.5 point :** • 0.25 • 0.25
Saisie de Xd et Yd:	**0.5 point** = 0.25+0.25

Saisie de Tx et Ty:
• Initialisation (lecture du premier élément)
• Parcours
• lecture d'un élément de Tx et un élément de Ty
• Contrôle de saisie
  ○ Testd'existence + Traitement de Existe
  ○ Test de différence avec Xd et Yd
	2.5 points :
	• 0.25
	• 0.5
	• 0.25+0.25
	• 0.25+0.75
	• 0.25

Remplissage de Td: • parcours • calcul	**1 point :** • 0.5 • 0.5

Tri :
• Parcours
• Comparaison
• permutation
  ○ dans Tx
  ○ dans Ty
  ○ dans Td
	3 points :
	• 1= 0.5 + 0.5
	• 0.5
	• 0.5
	• 0.5
	• 0.5

Affichage : • Parcours • Ecriture	**1.5 points** • 0.5 • 1= 0.5+0.5
TDNT **TDOG** **TDOL**	**0.25 point** **0. 5 point** **0.25 point**

Devoirs et examens sur : www.Kiteb.net

REPUBLIQUE TUNISIENNE MINISTRÉPUBLIQUE DE L'ÉDUCATION **EXAMEN DU BACCALAURÉAT** SESSION **2015**	Épreuve : **Informatique** Durée : 1 H 30 mn Coefficient : 0,5
Sections : **Mathématiques, Sciences expérimentales et Sciences Techniques**	

Le sujet comporte 4 pages numérotées de 1/4 à 4/4.
Les réponses à l'exercice 1 et 2 doivent être rédigées sur cette même feuille
qui doit être remise à la fin de l'épreuve

Exercice 1 (3,75 points)

Afin de réaliser les tâches décrites dans la première colonne du tableau suivant, un élève fournit les propositions suivantes. Remplir la colonne "Correction" en apportant les corrections nécessaires pour que ces propositions soient les plus adéquates relativement au choix de la structure itérative.

Tâche	Proposition	Correction
Saisir un entier positif n	n= [n= donnée ("Saisir un entier positif :")] Tant que (n□0) faire n= donnée ("Saisir un entier positif :") Fin tant que	....
Chercher la valeur maximale dans un tableau T de taille n.	Max = [i ← 1, Max ← T[1]] Répéter [ ] Si (T [i]> Max) alors Max ←T[i] Fin Si I←i+1 Jusqu'à (i > n)	....
Vérifier l'existence d'un caractère C dans un tableau T de n caractères.	Trouve = [Trouve □faux] Pour i de 1 à n faire [ ]Si (T[i] = C) Alors Trouve □Vrai Fin si Fin pour	....

Exercice 2 (5,25 points)

Soit U0 un entier naturel de quatre chiffres. A l'aide de ses quatre chiffres, on compose le plus grand entier et le plus petit entier formés par ces chiffres.

La différence de ces deux nombres donne U1, qui sera soumis au même traitement pour donner U2, etc. Jusqu'à ce que la suite U devienne stationnaire, c'est-à-dire, à un certain terme elle devient constante (ne change plus de valeur).

Soit l'algorithme suivant nommé Suite et permettant de déterminer les termes d'une suite U ayant comme premier terme U0, de les ranger dans un tableau T et de l'afficher (avec Max et Min sont deux modules qui déterminent respectivement le plus grand entier et le plus petit entier formés à partir des chiffres de Ui avec i >0).

0- Début Suite 1- Répéter Lire(U0) Jusqu'à (U0 ≥ 1000) et (U0 ≤ 9999) 2- i ←1 T[1]←U0	Répéter i←i+1 T[i] ← FN Max(U0) – FN Min(U0) U0← T[i] Jusqu'à (T[i] = T[i-1]) 3- Proc Afficher (T, i) 4- Fin Suite

Travail demandé :

Pour chacune des questions suivantes, cocher la ou les bonnes réponses.

1- Par quel appel peut-on remplacer la séquence 1 de l'algorithme Suite ?

☐ Proc Saisir (N) ☐ Proc Saisir (U0)

☐ Procédure Saisir (Var N : entier) ☐ U0← Proc saisir (N)

2- Quels sont les en-têtes qui correspondent à la déclaration de la procédure Afficher ?

☐ DEF Proc Afficher (Var T : tab)

☐ DEF Proc Afficher (T : tab ; N : entier)

☐ DEF Proc Afficher (i : entier ; T : tab)

☐ DEF Proc Afficher (T[i] : entier)

3- L'en-tête suivant de la fonction Max est erroné : DEF FN Max (X : entier) Quel est l'origine de l'erreur ?

☐ Le mode de passage des paramètres est erroné.

☐ Le nom du paramètre effectif est différent du nom du paramètre formel.

☐ Le type du résultat est manquant.

☐ Le type du paramètre effectif est incompatible avec celui du paramètre formel.

4- Si on veut remplacer la séquence 2 par l'appel d'un module :

a. Quelle sera sa nature ?

☐ Une procédure ☐ Une fonction

b. Quels seront les paramètres effectifs à utiliser ?

☐ T, i et U0 ☐ T[i] et U0

☐ T et U0

☐ T et i

5- Quel sera le tableau de déclaration des objets de l'algorithme Suite ?

☐ T.D.O.G ☐ T.D.O.G

Objet	Type
T	Tab
U0	Entier

Objet	Type
T	Tab
I, U0	Entier
Max, Min	Fonction
Afficher	Procédure

6- Pour U0 égale à 5360, quel sera le résultat de l'affichage de l'algorithme Suite ?

T

5843	5085	7992	7173	6354	3087	8352	6147	6174

T

5843	5085	2970	6930	5940	4950	4950

Problème (11 points)

Un nombre M est dit « nombre premier sûr », s'il est un nombre premier de la forme 2*p + 1 avec p un nombre premier.

Exemples :

- ✓ Si M = 11, alors M est un nombre premier sûr. En effet, 11 est premier et il peut s'écrire sous la forme 2*p+1 où p = 5 qui est un nombre premier.
- ✓ Si M = 31, alors M n'est pas un nombre premier sûr. En effet, 31 est premier et il peut s'écrire sous la forme 2*p+1 où p = 15 qui n'est pas un nombre premier.

NB : Un nombre entier supérieur à 1 est dit premier s'il n'est divisible que par 1 et par lui-même.

On se propose d'écrire un programme qui permet de :

1. Remplir un tableau T par N entiers strictement supérieurs à 1 (avec $10 \leq N < 45$).
2. Trier dans l'ordre croissant les éléments premiers sûrs du tableau T suivis du reste des éléments sans tri.
3. Afficher le tableau T résultant.

Exemple : Pour N = 10 et le tableau T suivant :

le contenu du tableau suivant :

T	5	25	59	23	13	47	31	100	7	107
	1	2	3	4	5	6	7	8	9	10

Le programme affichera

T	5	7	23	47	59	107	25	13	31	100
1	2	3	4	5	6	7	8	9	10	

Eléments premiers sûrs triés dans un ordre croissant

Eléments non premiers sûrs

Travail demandé :

1) Analyser le problème en le décomposant en modules.

2) Analyser chacun des modules envisagés.

Bac 2015
Corrigé : Informatique

Sections : Math, Techniques et Sciences Expérimentales

Exercice 1 (1.25 x 3 = 3.75 points)

Afin de réaliser les tâches décrites dans la première colonne du tableau suivant, un élève fournit les propositions suivantes. Apporter les corrections nécessaires pour que ces propositions soient les plus adéquates relativement au choix de la structure itérative.

Tâche	Proposition	Correction
Saisir un entier **n** positif.	n= [n= donnée ("Saisir un entier positif :")] Tant que (n<0) faire n= donnée ("Saisir un entier positif :") Fin tant que	**n=[ ]** **Répéter** **n= donnée ("Saisir un entier positif :")** **Jusqu'à (n ≥ 0)**
Chercher la valeur maximale dans un tableau **T** de taille **n**.	Max = [i ← 1, Max← t[1]] Répéter [ ] Si (T [i]> Max) alors Max ←T[i] Fin Si i←i+1 Jusqu'à (i > n)	**Max = [Max← T[1]]** **Pour i de 2 à n faire** **Si (T[i]> Max) alors** **Max ←T[i]** **Fin Si** **Fin pour**
Vérifier l'existence d'un caractère **C** dans un tableau **T** de **n** caractères.	Trouve= [Trouve ←faux] Pour i de 1 à n faire [ ] Si (T[i] = C) Alors Trouve ←Vrai Fin si Fin pour	**Trouve= [i ← 0]** **Répéter** **i←i+1** **Trouve ← T[i] = C** **Jusqu'à (Trouve) ou (i= n)**

Exercice 2 (0.75 x 7 = 5.25 points)

Pour chacune des questions suivantes, cochez la ou les bonnes réponses.

1- Par quel appel peut-on remplacer la séquence 1 de cet algorithme ?

- [ ] Proc Saisir (N)
- [ ] Procedure Saisir (Var N : entier)
- [x] Proc Saisir (U0)
- [ ] U0← Proc saisir (N)

2- Quelles sont les entêtes qui correspondent à la procédure **Afficher**?

- [ ] DEF Proc Afficher (Var T : tab)
- [x] DEF Proc Afficher (T : tab ; N : entier)
- [ ] DEF Proc Afficher (i : entier ; T : tab)
- [ ] DEF Proc Afficher (T[i] : entier)

3- L'entête suivante de la fonction Max est erronée : DEF FN **Max** (X : entier)
Quel est l'origine de l'erreur ?

- [ ] Le mode de passage des paramètres est erroné.
- [ ] Le nom du paramètre effectif est différent du nom du paramètre formel.
- [x] Le type du résultat manque.
- [ ] Le type du paramètre effectif est incompatible avec celui du paramètre formel.

4- Si on veut remplacer la séquence 2 par l'appel d'un module :

a. Quelle sera sa nature ?

- [x] Procédure
- [ ] Fonction

b. Quels seront les paramètres à utiliser ?

- [x] T, i et U0
- [ ] T[i] et U0
- [ ] T et U0
- [ ] T et i

5- Quel sera le tableau de déclaration des objets de l'algorithme Suite ?

[ ]

Objet	Type
T	Tab
U0	Entier

[x]

Objet	Type
T	Tab
I, U0	Entier
Max, Min	Fonction
Afficher	Procédure

6- Pour une valeur donnée d'**U0** égale à **5360.** Quel sera le résultat de l'affichage de l'algorithme **Suite** ?

[x] **T**

5843	5085	7992	7173	6354	3087	8352	6147	6174

[ ] **T**

5843	5085	2970	6930	5940	4950	4950

Problème (11 points)

TDNT

Type
Vect = tableau de 44 entiers

a) ANALYSE DU PROGRAMME PRINCIPAL

Résultat = PROC **afficher** (T, n)
T = PROC **trier** (T, nb)
T, nb = PROC **ranger** (T, n, nb)
T, n = PROC **remplir** (T, n)

TDOG

Objet	T/N
T	Vect
n, nb	Entier
Remplir	Procédure
Ranger	Procédure
Trier	Procédure
Afficher	Procédure

b) ANALYSE DE LA PROCEDURE REMPLIR

```
DEF PROC remplir (var t : vect ; var n : entier)
Résultat = t, n
t= [ ]
Pour i de 1 à n faire
    Répéter
      t[i]= donnée ("saisir un entier :")
    Jusqu'à (t [i]>1)
Fin pour
n= [ ]
Répéter
  n= donnée ("saisir un entier :")
Jusqu'à (n dans [10..44])
```

Objet	Type
i	Entier

c) ANALYSE DE LA PROCEDURE RANGER

```
DEF PROC Ranger (Var t : vect ; n : entier ; Var nb: entier)
Résultat = t, nb
    (t, nb) = [nb←0]
Pour i de 1 à n faire
  Si (FN Premier (t[i]) et (FN Premier ((t[i]-1) div 2) alors
        nb← nb+1
        aux← t[i]
        t[i] ← t[nb]
        t[nb] ← aux
   Fin si
Fin pour
Fin Ranger
```

Objet	Type
Premier	Fonction
PremSur	Fonction
i, j	Entier
aux	Entier

d)ANALYSE DE LA FONCTION

DEF FN Premier (r : entier) : booléen
Résultat = premier ← (r mod d=0)
d = [d ←1]
Répéter
d←d+1
Jusqu'à (r mod d =0) ou (d > racinecarré(r))
Fin premier

Objet	Type
d	Entier

e) ANALYSE DE LA PROCEDURE TRIER

Def Proc TRIER (VAR T : VECT ; N : ENTIER)

Résultat = T
T = [ ] Répéter
Echange ← faux
Pour i de 1 à n-1 faire
Si (T[i] > T[i+1]) alors
Aux←T[i]
T[i]←T[i+1]
T[i+1]←Aux
Echange ← vrai
Fin si
Fin pour
n←n-1
Jusqu'à (n=1) ou (Echange = faux)

Objet	Type
Aux	Entier
i	Entier
Echange	booléen